

# smesh User's Guide

*K. K. Chand*

**U.S. Department of Energy**

Lawrence  
Livermore  
National  
Laboratory

**May 5, 2003**

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

## **DISCLAIMER**

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This research was supported under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

# smesh User's Guide

A 2D unstructured mesh generator for Overture

Kyle K. Chand Centre for Applied Scientific Computing  
Lawrence Livermore National Laboratory  
Livermore, CA, 94551  
chand1@llnl.gov  
<http://www.llnl.gov/casc/Overture>

May 5, 2003

## Abstract

**smesh** is a general purpose, interactive, 2D unstructured mesh generator based on **Overture**. It supports three kinds of mesh generation techniques : structured patches with transfinite interpolation (TFI); unstructured triangles based on an advancing front technique; and a Cartesian cutcell/triangle hybrid method. Meshes are generated in a generalized “mulit-block” manner where each “block”, or region, can be one of the three mesh types. Geometry definitions can be created interactively by placing points and interpolating curves. Spacing information is provided by both the curve discretization (which can be stretched) and a user specified preferred grid spacing for a region. A mesh optimization procedure is available for the non-TFI regions for mesh quality improvement. Each mesh region is given an unique identifier and an optional string name. Meshes are exported to a modified “ingrid” format including mesh region identifiers and names. Facilities for command scripting and batch running are available.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>How To...</b>	<b>5</b>
2.1	Create geometry (points and curves) . . . . .	5
2.2	Specify grid points and stretching on curves . . . . .	5
2.3	Create a TFI region . . . . .	7
2.4	Create an unstructured region . . . . .	7
2.5	Output the mesh to a file . . . . .	8

<b>3</b>	<b>Examples</b>	<b>8</b>
3.1	A simple TFI region . . . . .	8
3.1.1	command file . . . . .	8
3.2	A simple unstructured region . . . . .	9
3.2.1	command file . . . . .	10
3.3	An example with multiple regions . . . . .	11
3.3.1	command file . . . . .	11
3.4	Stretched regions . . . . .	14
3.4.1	command file . . . . .	14

# 1 Introduction

**smesh** can be used to generate two dimensional unstructured meshes. It can be used interactively, as in Figure 1, or without graphics in a batch mode using scripted commands. The program can be invoked on the command line using:

```
smesh [noplot] [nopause] [command_file]
```

When started, **smesh** will either process a command file or, if a file is not specified, begin an interactive session. The optional command file is a script of **smesh** commands; this script is usually adapted from the log of a previous interactive session. **smesh** stores all the commands executed during a run (interactive or batch) in a file called **smesh.cmd**. This file can be given to **smesh** as a command file to reproduce the interactive session. **noplot** starts **smesh** without graphics; the code will then be driven by text input. When **nopause** is specified, “pause” commands in the command file are ignored.

Generating a mesh with **smesh** proceeds with the following steps:

- create points and curves
- optionally choose point distributions on curves
- use a collection of curves to define and generate a region
- make more geometry/regions
- output the mesh to a text file

Figure 2 shows the main interface used to perform these tasks. Clicking the “Create Curves” button brings up the curve generation interface discussed in Section 2.1. A variety of operations can be performed using mouse clicks, including: deletion of curves and regions; specifying curve

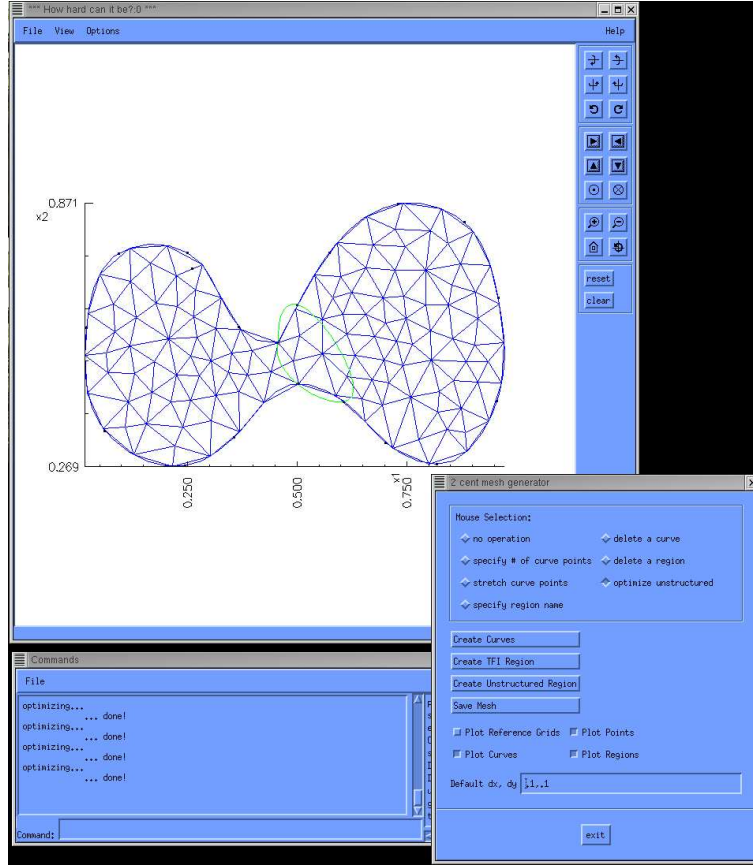


Figure 1: Screenshot of **smesh** in action

point density and stretching; specifying text names for regions; and directing the code to optimize a specific unstructured region. The region creation buttons bring up “wizard” like interfaces for the generation of transfinite interpolation (TFI) or unstructured regions using a collection of the already defined curves. “Save Mesh” allows the user to save the grid in the text format described in Section 2.5. Toggle buttons are used to control the plotting of various pieces of information. For example, “Plot Reference Grids” will plot Cartesian grids bounding each region with the nominal target grid spacing for each respective region. Finally, a text box at the bottom of the window can be used to specify the default grid spacing for new regions (note this can be overridden when a region is created).

Figure 3 illustrates the three kinds of unstructured mesh generation supported by **smesh**. Each one of these mesh regions begins with the specification of an outer bounding curve (inner curves

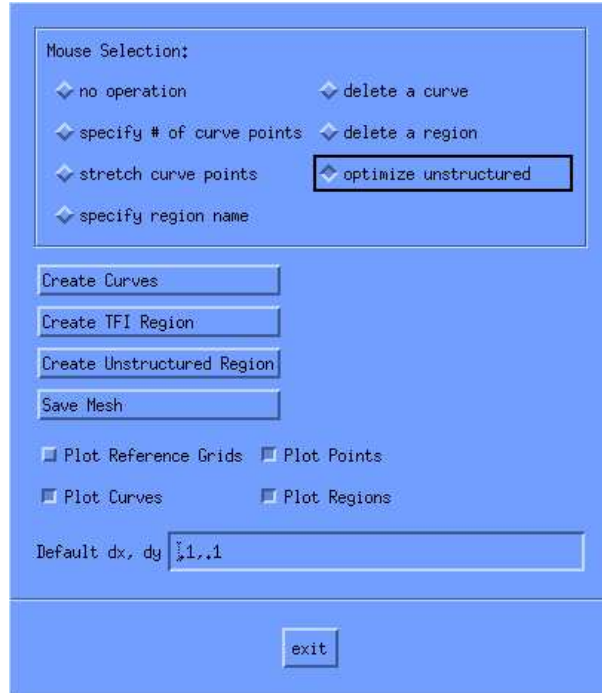


Figure 2: The main command window

are also possible for non-TFI regions). The “cutcell hybrid” method first covers the area bounded by the curve with a uniform Cartesian grid with a user defined spacing. This Cartesian grid is then cut by the bounding curve; any grid outside the curve is discarded. An advancing front mesh generator connects the curve to the Cartesian grid with triangles forming the completed mesh. The unstructured triangulation in the middle of Figure 3 gives the discretized outer curve to an advancing front mesh generator which fills the region with triangles. Finally, the TFI region begins with four curves that specify the outer boundary and then uses transfinite interpolation to compute the mesh points on the interior of the domain. These methods are described in more detail in Sections 2.3, and 2.4.

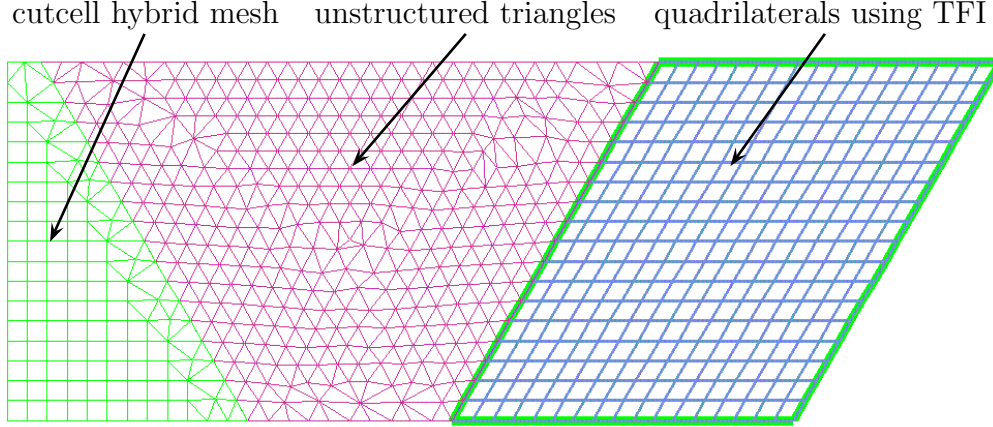


Figure 3: The three types of unstructured mesh generation supported by `smesh`

## 2 How To...

### 2.1 Create geometry (points and curves)

Geometry used as the bounding curves of mesh regions are created using the interface shown in Figure 4. Points can be placed either by interactively clicking with the mouse (after selecting the “Build Point” mouse mode toggle) or by inputting the coordinates in the text box at the bottom. Once at least two points are placed, a curve can be created by changing the mouse mode to “Interpolate Curve” and then picking a sequence of points with the mouse. After the last point has been chosen, clicking “Done” at the bottom of the graphics window creates a spline curve that interpolates the selected points. At this point, the curve could be edited using mouse modes such as “Split Curve”; new curves can be created; or the user can exit the geometry creation interface.

### 2.2 Specify grid points and stretching on curves

Once curves have been created, it is often useful to specify the number of points on the curve and how points should be distributed. The main interface (Figure 2) mouse mode “specify number of curve points” enables the specification of points on the curve using the mouse. Clicking on a curve will bring up a small dialog box with on text entry specifying the number of points. Typically, using “auto” in the text box will instruct `smesh` to automatically determine the spacing given the



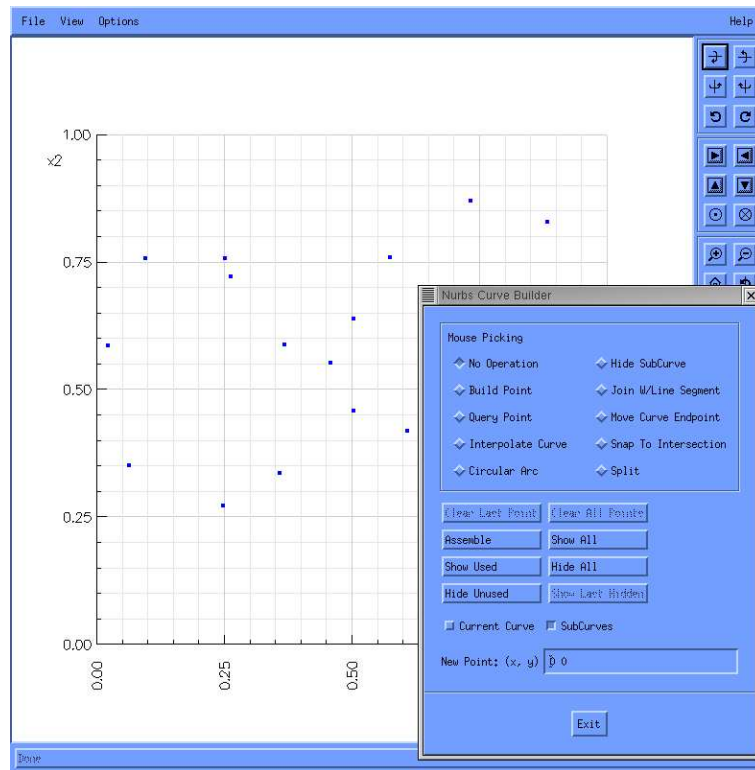


Figure 4: Geometry creation interface

requested mesh size of a region using the curve. Inputting a number greater than zero will force that number of points to be used on the curve. Note that once a curve has been used by a region its point distribution cannot be changed.

Specifying stretching occurs in a similar manner to point size; one selects the “stretch curve points” as the mouse mode and then clicks on the curve to alter. Clicking on a curve will bring up *Overture*’s stretching interface, described in more detail in *Overture*’s documentation [1].

## 2.3 Create a TFI region

Structured regions based on TFI are created by selecting four curves that bound the region; these curve will be referred to as the “left”, “right”, “bottom” and “top”, naturally. When “Create TFI Region” is selected, a dialog window appears that asks the user to select the “left” curve. Adjacent curves can be concatenated to form longer and more complex boundaries that may be adjacent to more than one other region. Selecting adjacent curves will automatically concatenate them and display the composite curve in bold black. When the curve is complete, selecting “done” in the dialog window proceeds to the next curve. Each curve is selected as described in the order specified above. Once all four curves are selected, another dialog appears that allows the user to specify a text name for the region and adjust the requested mesh spacing. Selecting “Generate” at the bottom of this dialog box generates the mesh and returns to the main interface.

## 2.4 Create an unstructured region

An unstructured region is created by selecting one outer bounding curve and optional inner curves. When “Create Unstructured Region” is selected, a dialog window requesting the user to choose the outer bounding curve pops up. Curve selection proceeds in the same manner as a TFI region; compound curves are selected by clicking on adjacent curves and finishing by clicking “done” on the popup. Again, the curve under construction is displayed in bold black as curves are selected. Once the outer curve has been chosen, a new dialog will request an inner curve. Inner curves are selected in the same manner as the outer curves. When all the inner curves are selected, clicking “done” in the dialog without choosing a curve will present a window in which the user can specify a name, change the spacing or select between the Cartesian cutout hybrid and all triangular mesh generation techniques. As before, pushing “Generate” creates the mesh and returns to the main interface. An unstructured mesh may be optimized (have vertices moved to more optimal locations) by selecting “optimize unstructured” as the mouse mode in the main interface. Simply clicking on an unstructured region will perform one iteration of a mesh optimization procedure on the region.

## 2.5 Output the mesh to a file

Clicking the “Save Mesh” button on the main interface will present a file selection dialog. Type or choose a filename for the mesh to write the data to a text file. At the end of each mesh file, a description of the file format is presented. The file format is a modified version of the “ingrid” text file:

```
FORMAT SUMMARY ::
element vertices specified counter-clockwise,
1 based index with 0 specifying a null vertex (vertex 4 of a triangle)
line 1 : Comment Line
line 2 : nRegions nVertices nElements maxNVertsInElement domainDimension rangeDimension
line 3 : 0 x0 y0
line 4 : 1 x1 y1
...
line nVertices+2 : nVertices-1 xNv yNv
line nVertices+3 : 0 reg0 e0v1 e0v2 e0v3 e0v4
line nVertices+4 : 1 reg1 e1v1 e1v2 e1v3
...
line nVertices+2+nElements : nElements-1 regNe eNv1 eNv2 eNv3 eNv4
line nVertices+3+nElements : region0ID region0Name
...
line nVertices+3+nElements+nRegions : regionN regionNName
This Summary
```

## 3 Examples

### 3.1 A simple TFI region

A TFI region is built using four curves. In this example, the curves are built as straight lines connecting four vertices. Figure 5 illustrates the process of constructing this type of region.

#### 3.1.1 command file

```
Create Curves
Mouse Mode Build Point
new point 2.055071e-01 6.812946e-01
new point 7.907053e-01 7.342878e-01
```

```

new point 2.698978e-01 1.025259e-01
new point 8.664590e-01 1.029423e-01
Mouse Mode Interpolate Curve
point for interpolation 1
point for interpolation 0
stop picking
point for interpolation 0
point for interpolation 2
stop picking
point for interpolation 2
point for interpolation 3
stop picking
point for interpolation 3
point for interpolation 1
stop picking
exit
Create TFI Region
select left 1
Done
select right 3
Done
select bottom 2
Done
select top 0
Done
generate
tb Plot Reference Grids 0
Save Mesh
/home/chand/overture/views/tools/2dmesh/tfi.msh
exit

```

## 3.2 A simple unstructured region

Unstructured regions also begin by defining a set of curves that define a connected domain. At least one curve is required to define the outer boundary. Multiply connected domains may be created by creating one or more inner boundary curves. There are two types of unstructured meshing techniques available. One simply fills the domain with triangles using *Overture*'s advancing front mesh generator. The second method builds a mixed triangle-quadrilateral mesh by cutting a Cartesian

grid with the bounding curves and then connecting the curves to the grid with triangles. These options are show in Figure 6.

### 3.2.1 command file

Create Curves

Mouse Mode Build Point

new point 1.411164e-01 5.980980e-01

new point 3.456383e-01 7.086174e-01

new point 6.107900e-01 6.916550e-01

new point 8.134314e-01 5.745974e-01

new point 8.210067e-01 4.336066e-01

new point 5.312484e-01 2.322357e-01

new point 6.915023e-02 2.534473e-01

Mouse Mode Interpolate Curve

point for interpolation 0

point for interpolation 1

point for interpolation 2

point for interpolation 3

point for interpolation 4

point for interpolation 5

point for interpolation 6

point for interpolation 0

stop picking

Mouse Mode Build Point

new point 3.352156e-01 4.913438e-01

new point 4.573885e-01 5.203880e-01

new point 4.687516e-01 4.123340e-01

new point 3.399701e-01 3.757012e-01

Mouse Mode Interpolate Curve

point for interpolation 7

point for interpolation 8

point for interpolation 9

point for interpolation 10

point for interpolation 7

stop picking

exit

dxdy .02,.02

```

Create Unstructured Region
select outer 0
Done
select inner 1
Done
Done
generate
tb Plot Reference Grids 0
pause
mm delRegion
delete region 0
Create Unstructured Region
select outer 0
Done
select inner 1
Done
Done
use cutout 0
generate
mm optimize
optimize region 1
optimize region 1
optimize region 1
optimize region 1

```

### 3.3 An example with multiple regions

This example demonstrates the creation of a mesh with multiple regions. Figure 7 shows three regions created using each of the mesh generation methods available. The TFI region at the top assembled its “bottom” curve from the two horizontal curves in the middle of the geometry. The region is adjacent to two unstructured regions that utilize the middle curves as subsets of their outer boundaries. Two circular arc curves form the inner boundary of an unstructured region at the lower left region of the mesh. Information about these regions is placed in the text file when the mesh is saved.

#### 3.3.1 command file

```
Create Curves
```

Mouse Mode Build Point  
new point 0 0  
new point 2 0  
new point 5 0  
new point 0 2  
new point 2 2  
new point 5 2  
new point 0 3  
new point 5 3  
new point 1 .75  
new point 1 1.25  
Mouse Mode Circular Arc  
arc segment 8 9  
radius of curvature .25  
arc segment 9 8  
radius of curvature .25  
point for interpolation 0  
point for interpolation 1  
stop picking  
point for interpolation 1  
point for interpolation 2  
stop picking  
point for interpolation 0  
point for interpolation 3  
stop picking  
point for interpolation 3  
point for interpolation 6  
stop picking  
point for interpolation 6  
point for interpolation 7  
stop picking  
point for interpolation 3  
point for interpolation 4  
stop picking  
point for interpolation 4  
point for interpolation 5  
stop picking  
point for interpolation 2

```

point for interpolation 5
stop picking
point for interpolation 5
point for interpolation 7
stop picking
point for interpolation 1
point for interpolation 4
stop picking
exit
dxdy .1,.1
Create TFI Region
select left 5
Done
select right 10
Done
select bottom 7
select bottom 8
Done
select top 6
Done
generate
tb Plot Reference Grids 0
Create Unstructured Region
select outer 4
select outer 2
select outer 11
select outer 7
Done
select inner 1
select inner 0
Done
Done
generate
Create Unstructured Region
select outer 11
select outer 3
select outer 9
select outer 8

```



```

Done
Done
use cutout 0
generate
mm optimize
optimize region 3
optimize region 3

```

## 3.4 Stretched regions

The distribution of points along curves can be adjusted by specifying the number of points using the main interface as well as stretching the points along a curve using **Overture** 's **StretchTransform** interface [2]. An example of both these tools is displayed in Figure 8 for both unstructured and TFI regions. Note, when a cutcell unstructured region is formed, the stretching information is not propagated to the interior.

### 3.4.1 command file

```

dxdy .05,.05
Create Curves
new point 0 0
new point .2 0
new point .2 1
new point 0 1
DISPLAY AXES:0 0
Mouse Mode Interpolate Curve
point for interpolation 0
point for interpolation 1
stop picking
point for interpolation 1
point for interpolation 2
stop picking
point for interpolation 2
point for interpolation 3
stop picking
point for interpolation 3
point for interpolation 0
stop picking

```

```

exit
DISPLAY AXES:0 0
mm nCurvePts
points on curve 0 11
points on curve 1 25
points on curve 3 25
mm strCurve
stretch points on curve 1
print grid statistics
stretch grid
Stretch r1:exp
STP:stretch r1 exp: min dx .02
stretch grid
STP:stretch r1 exp: cluster at r=0
stretch grid
pause
exit
stretch points on curve 3
Stretch r1:exp
STP:stretch r1 exp: min dx .02
STP:stretch r1 exp: cluster at r=1
stretch grid
pause
close r1 stretching parameters
exit
Create Unstructured Region
select outer 0
select outer 1
select outer 2
select outer 3
Done
Done
use cutout 0
name stretched
generate
tb Plot Reference Grids 0
mm optimize
optimize region 0

```

optimize region 0  
optimize region 0

## References

- [1] D. L. Brown, W. D. Henshaw, and D. J. Quinlan. Overture: Object-oriented tools for overset grid applications. Technical report, Lawrence Livermore National Laboratory, 1999. UCRL-JC-134018.
- [2] William D. Henshaw. *Mappings for Overture A Description of the Mapping Class and Documentation for Many Useful Mappings*. Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002. UCRL-MA-132239.

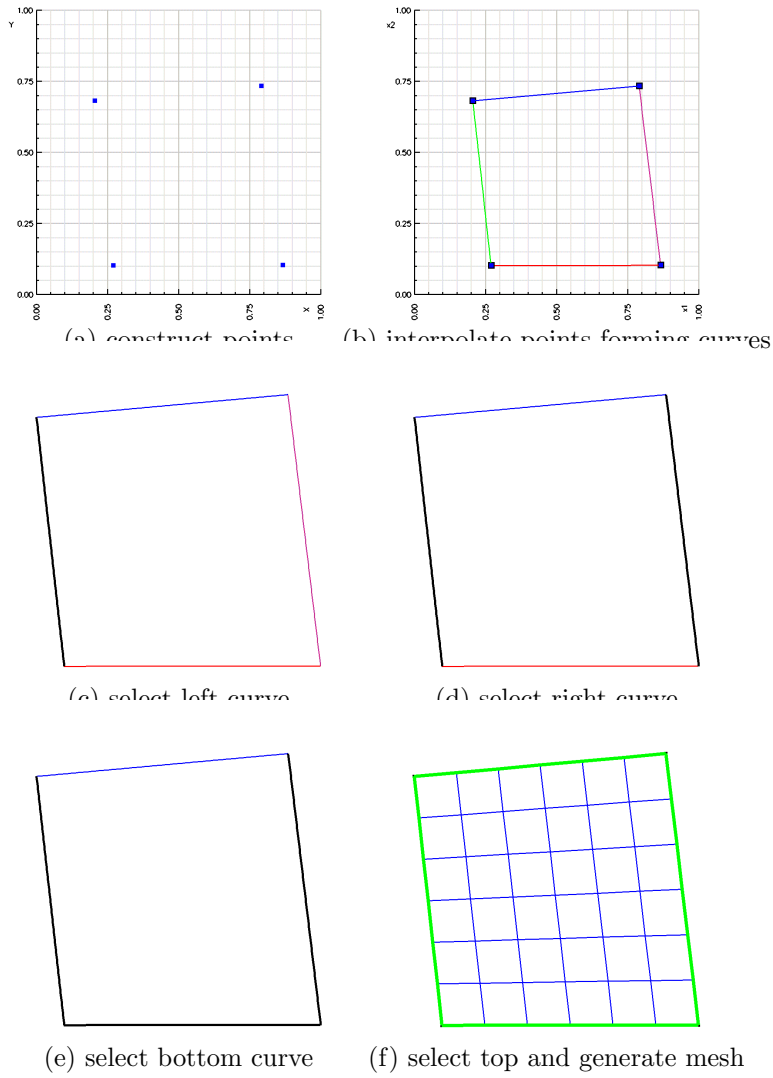


Figure 5: Construction of a tfi region: (a) construct points; (b) interpolate points to form bounding curves; (c)-(f) assemble four curves to form the grid (f).

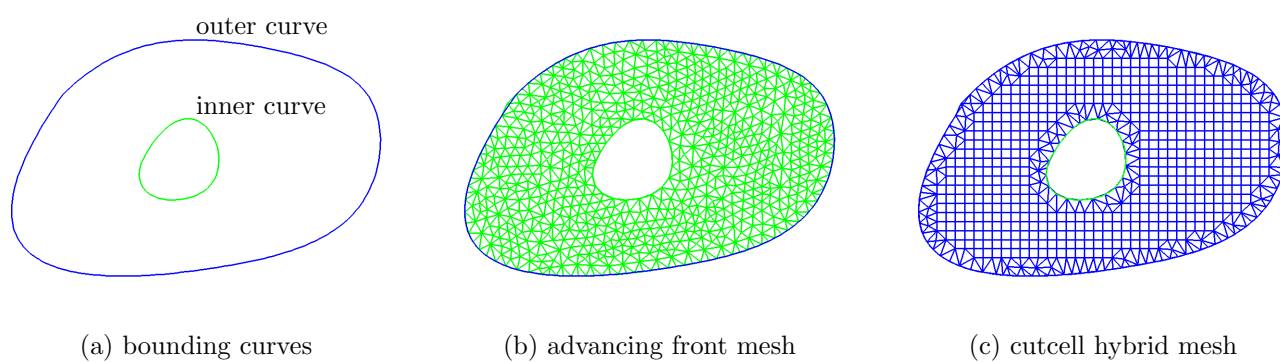


Figure 6: Unstructured regions are built using inner and outer bounding curves (a); they come in two flavors: (b) all triangles and (c) mixed quads and triangles

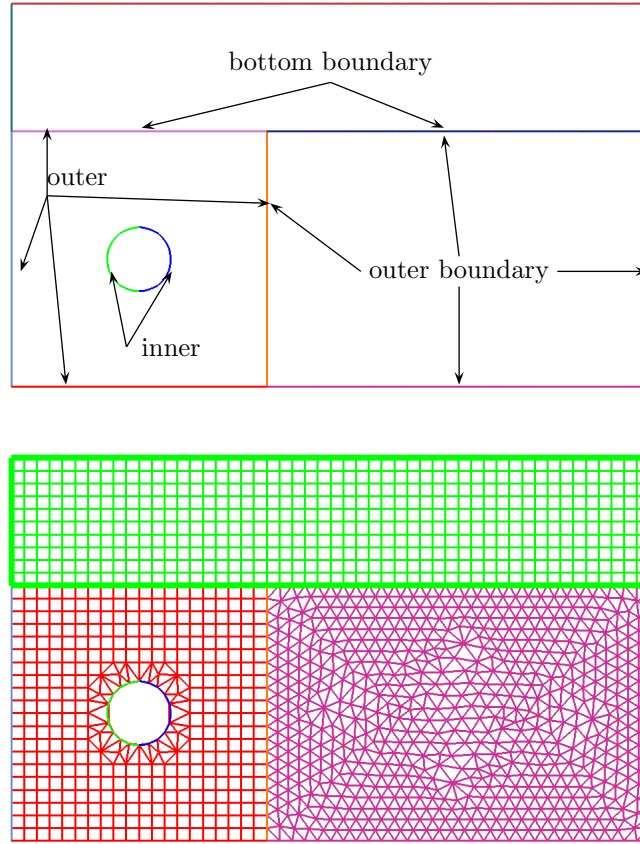


Figure 7: A multi-region mesh consisting of regions of each type; note that composite curves are used to form the “bottom” boundary of the top, TFI region, and the inner boundary of the lower left unstructured region.

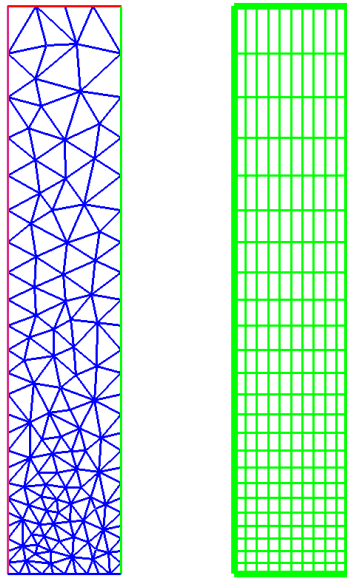


Figure 8: `smesh` can be used to generated graded meshes

Approved for public release; further dissemination unlimited



\_\_\_\_\_